
docker

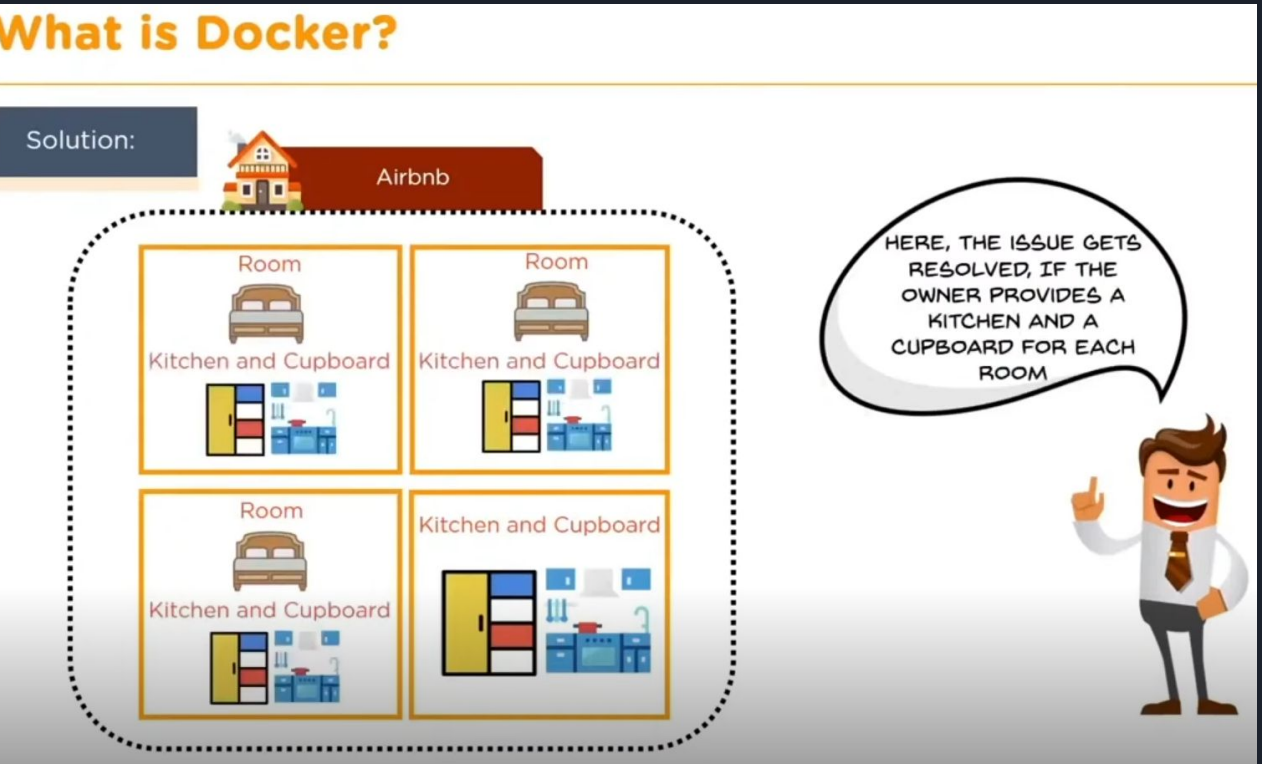
Before Docker

Considering
1. rooms = applications
2. kitchen = framework

If all of the three rooms members wants the kitchen setup different then it is impossible in this house.

Similarly, before docker users were not able to use different OS in same infrastructure.

In presence of Docker

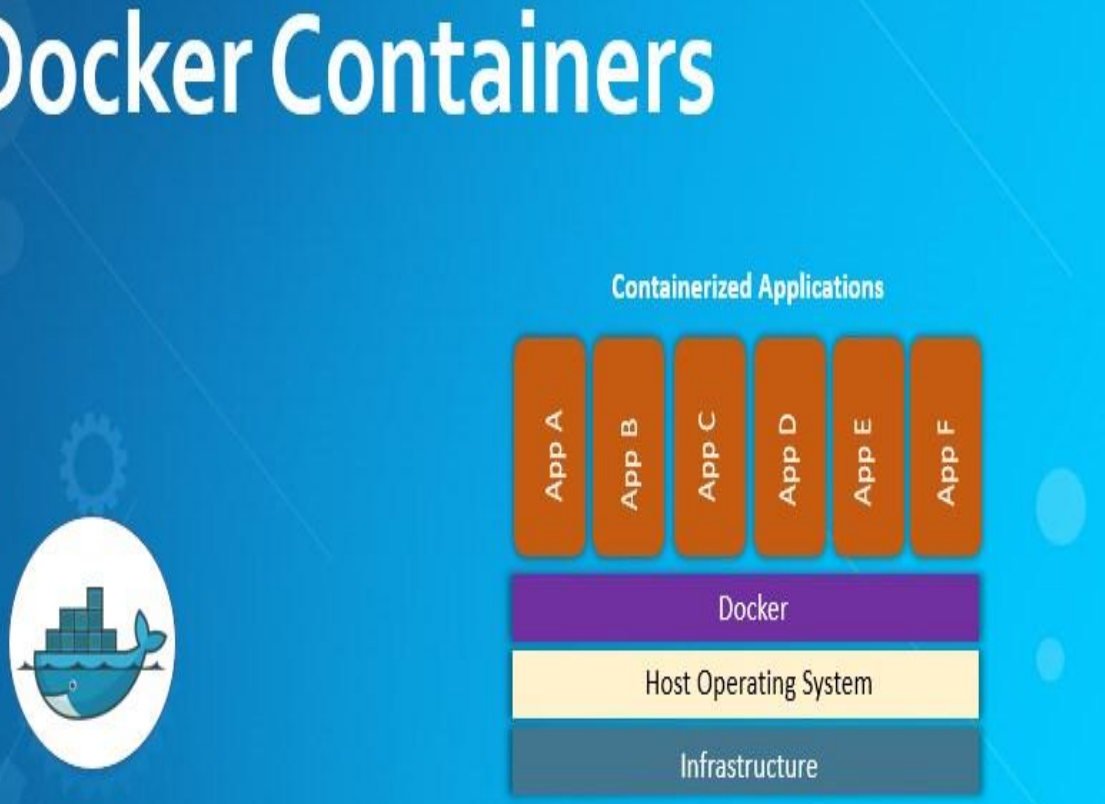


But in case of docker we can use various OS in a single infrastructure.

As the rooms are given separate kitchens we can have differenet frameworks for every software applications.

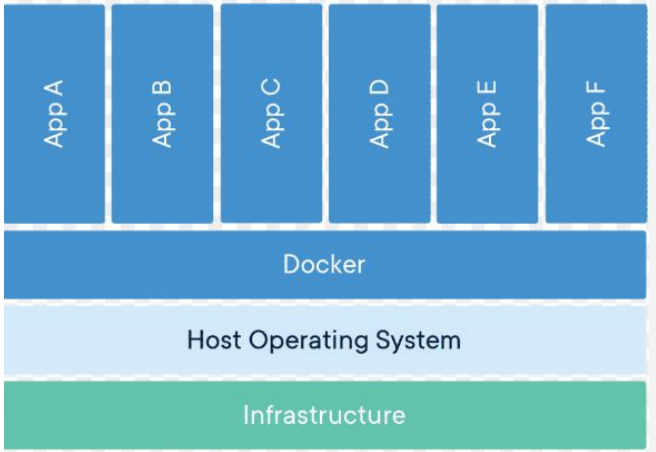
Introduction:

- Docker is a open-source containerization platform that is used to build, deploy and manage application container on common OS.
- Docker is basically using OS-level virtualization.
- Docker containers are build from docker images.



Features of Docker:

All of the container share same OS on the basis of their requirement.

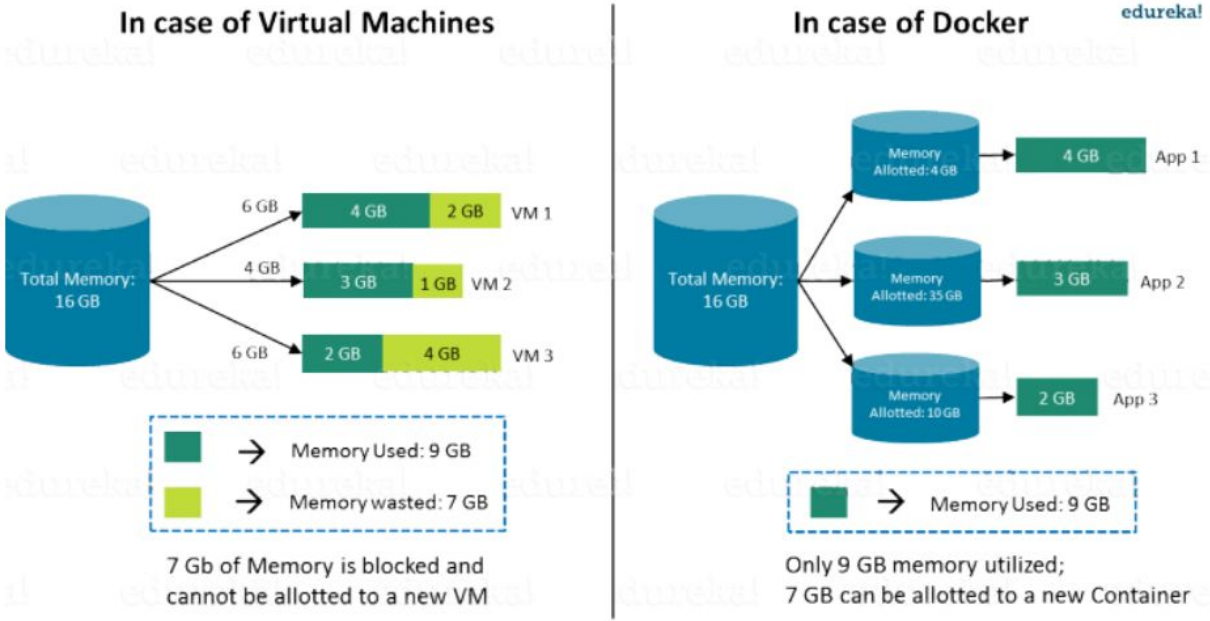


Docker is called lightweight since the container consists of OS but very small in size as compared to RAM size allocated. For eg: RAM size is 4Gb and OS size will be just about 60Mb.

Docker also allows us to have shared containers.

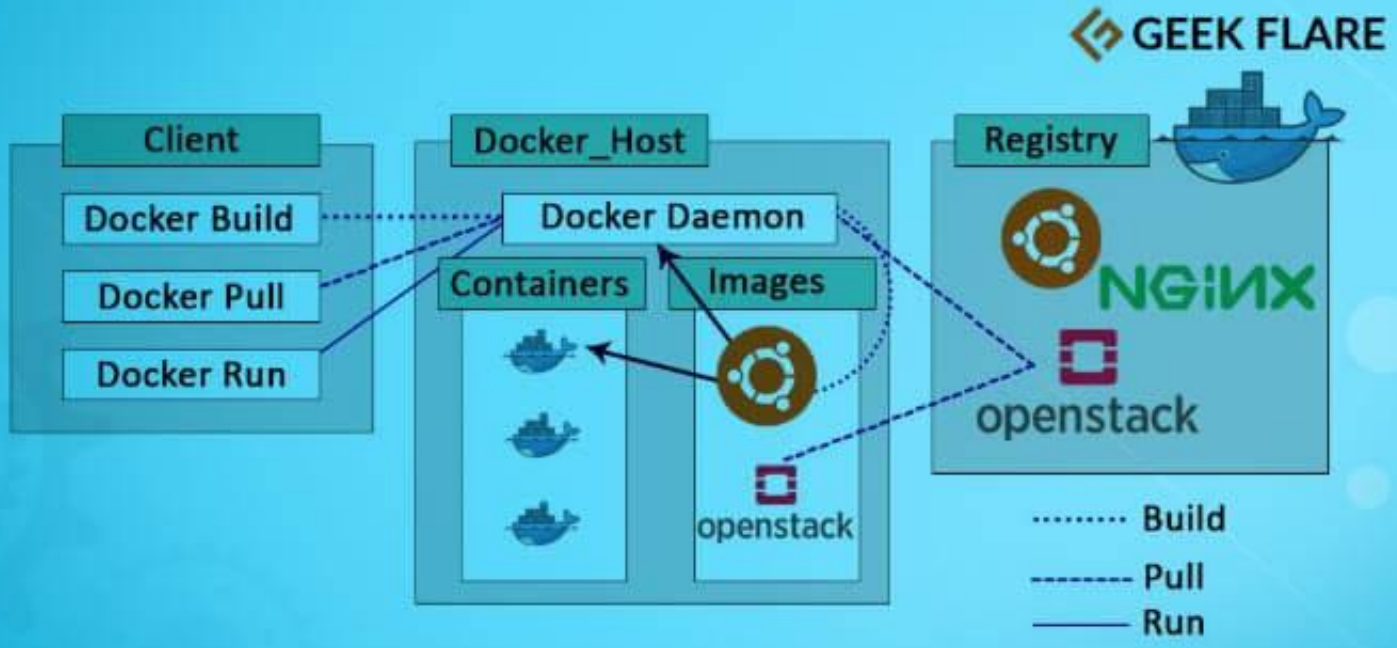
Features of Docker:

Docker is called lightweight since the container consists of OS but very small in size as compared to RAM size allocated. For eg: RAM size is 4Gb and OS size will be just about 60Mb.

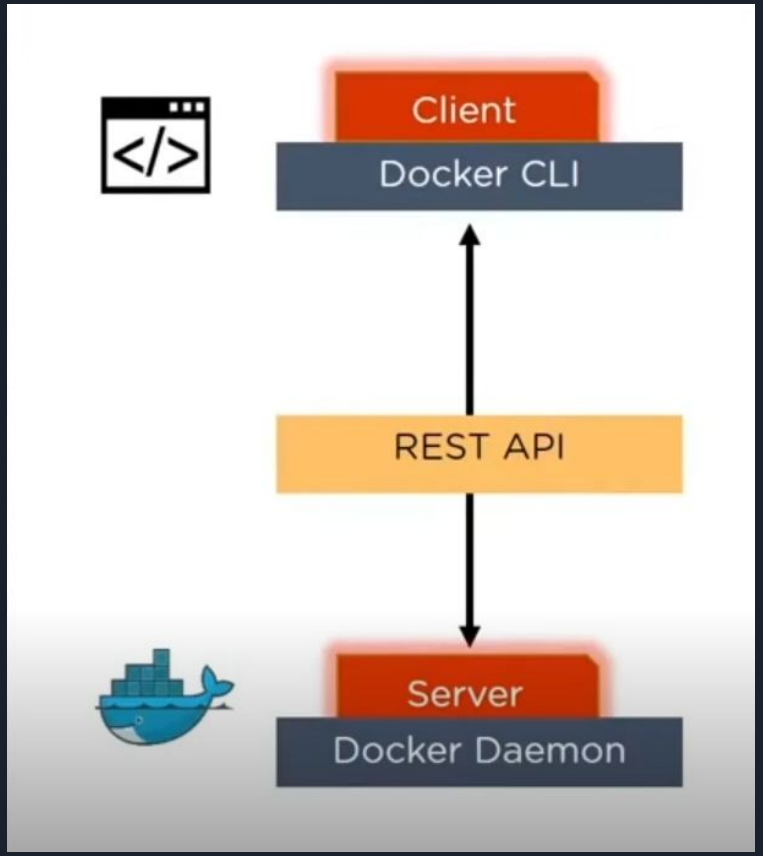


Container have flexibility to allocate the RAM size as per their need.

Docker Architecture



Description of architecture



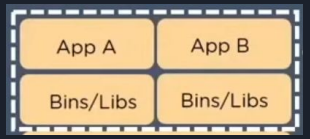
1. Docker uses a **client-server architecture.**
2. We can use **client and daemon** on same system or they **communicate using REST API.**
3. Docker registry is a storage to store the images. Docker hub is the public registry that can be used by anyone.
4. Image is a read-only template with instruction for creating container.
5. Container is a runnable instance of an image.

Components of Docker

Docker Client & server

Docker Images

Docker Registry



Docker Containers

1. Docker client talks to docker daemon. User passes commands to server to build images & run container. Docker daemon build, run, and distributes containers.
2. Docker image is a template with instruction, used to create docker container.
3. Docker image is stored in a docker hub or registry.
4. Container is a standalone, executable software package which includes applications & their dependencies. Containers are always isolated from each other.

What can we use Docker for?

1. Fast, consistent delivery of application.
2. Its container-based platform allows for highly portable workloads. Containers can run on a developer's local laptop, on VM in data center, on cloud providers, or in a mixture of environment.
3. We can use docker for high density environments and for small deployments where we need to do more with few resources, i.e. running more workloads on the same hardware.

1. docker images

- This command is used to show whether our docker contains any image or not.

2. docker search image_name

- This command is used to search for the available images.

3. docker pull img_nam

- This command is used to pull image from the docker registry(i.e. docker hub).

4. docker run image_name

- This command is used build the image and run the container.
- If we use:
  docker run -it img_name
  Then we can have shell access to container.

5. docker ps

- This command is used to show whether there is any running container or not.

- If -a is attached to the command then it shows all the containers.

6. docker start conta_id

- This command starts the stopped container.
- Similarly use stop top halt the running container.

7. docker commit -m “content” -a “author_name” conta_id user_name

- -m switch is for the commit message and -a is used to specify authors.
- When we commit an image, the new image created from existing image is saved locally on your computer.

8. Steps for pushing images into the docker hub

1. Login into docker hub

Command:

docker login -u user_name

2. Tag the image referenced by using name or ID to the repository.

Command:

docker tag img_nam/img_tag usr_nm/repository_nm:tag_name

3. Finally push the image to the repository.

Command:

docker push usr_nam/repo_nm

9. docker rename conta_id new_name

- This command is used to rename the docker.